

Dear Diary

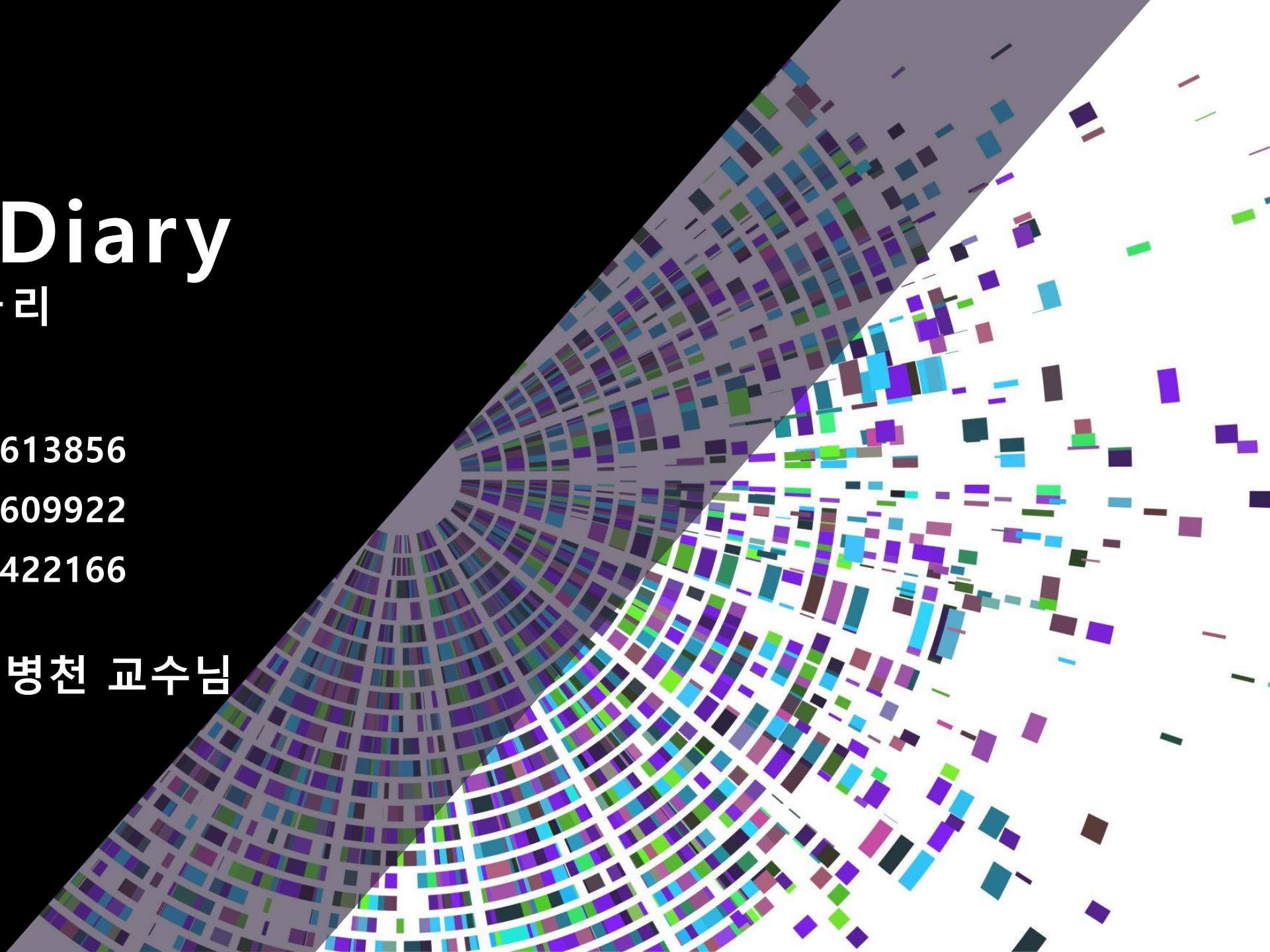
브로콜리

이경호 91613856

김지환 91609922

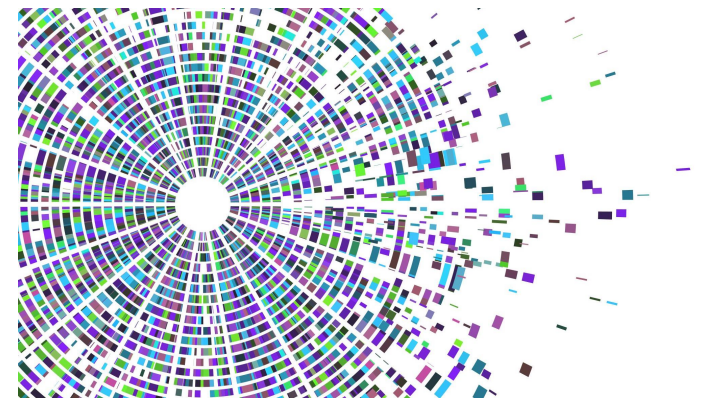
이지환 91422166

지도 교수 : 이병천 교수님



목차

- 구성
 - 개발 환경
 - 개발 내용 및 시스템 운영
 - 결론 및 기대효과
-



[구성]

회원가입 및 로그인

Facebook OAuth
Google OAuth
Firebase Login

Diary 작성

Firestore
permission: READ_EXTERNAL_STORAGE
Storage

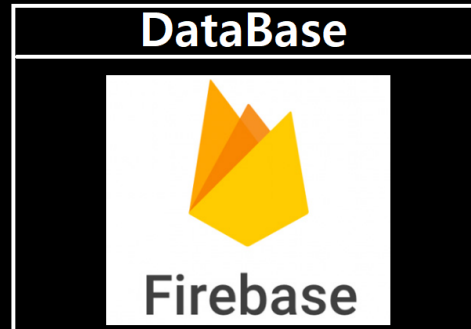
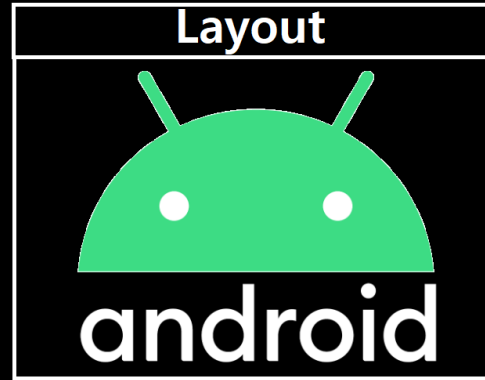
Comment

Firestore Database

알림

Fcm

[개발환경]



[개발 내용 및 시스템 운영 1/9]

Login

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(binding.root)  
    auth = FirebaseAuth.getInstance()  
  
    binding.emailLoginbutton.setOnClickListener {  
        signinAndSignup()  
    }  
    binding.facebookLoginButton.setOnClickListener {  
        facebookLogin()  
    }  
  
    binding.googleSignInButton.setOnClickListener {  
        googleLogin()  
    }  
}
```

Dear Diary

Email

Password

회원가입 및 로그인

다른 계정으로 로그인 하기



FACEBOOK 로그인



GOOGLE LOGIN

[개발 내용 및 시스템 운영 2/9]

main & detail

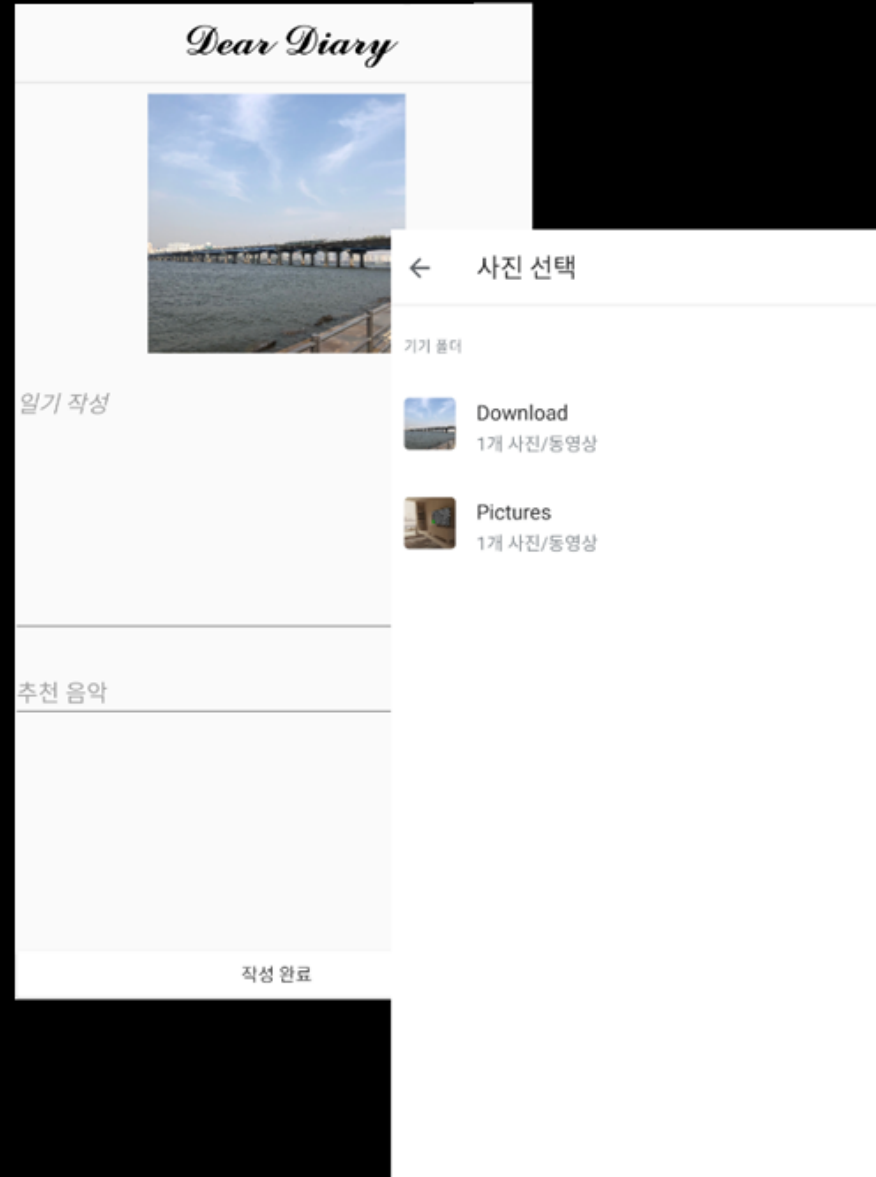
```
inner class DetailViewRecyclerViewAdapter : RecyclerView.Adapter<RecyclerView.ViewHolder>() {  
    var contentDTOs: ArrayList<ContentDTO> = arrayListOf()  
    var contentUidList: ArrayList<String> = arrayListOf()  
  
    init {  
        firestore?.collection("diary")?.orderBy("timestamp", Query.Direction.DESCEDING)  
            ?.addSnapshotListener { value, error ->  
                contentDTOs.clear()  
                contentUidList.clear()  
                if (value == null) return@addSnapshotListener  
                for (snapshot in value!!.documents) {  
                    var item = snapshot.toObject(ContentDTO::class.java)  
                    contentDTOs.add(item!!)  
                    contentUidList.add(snapshot.id)  
                }  
                notifyDataSetChanged()  
            }  
    }  
}
```



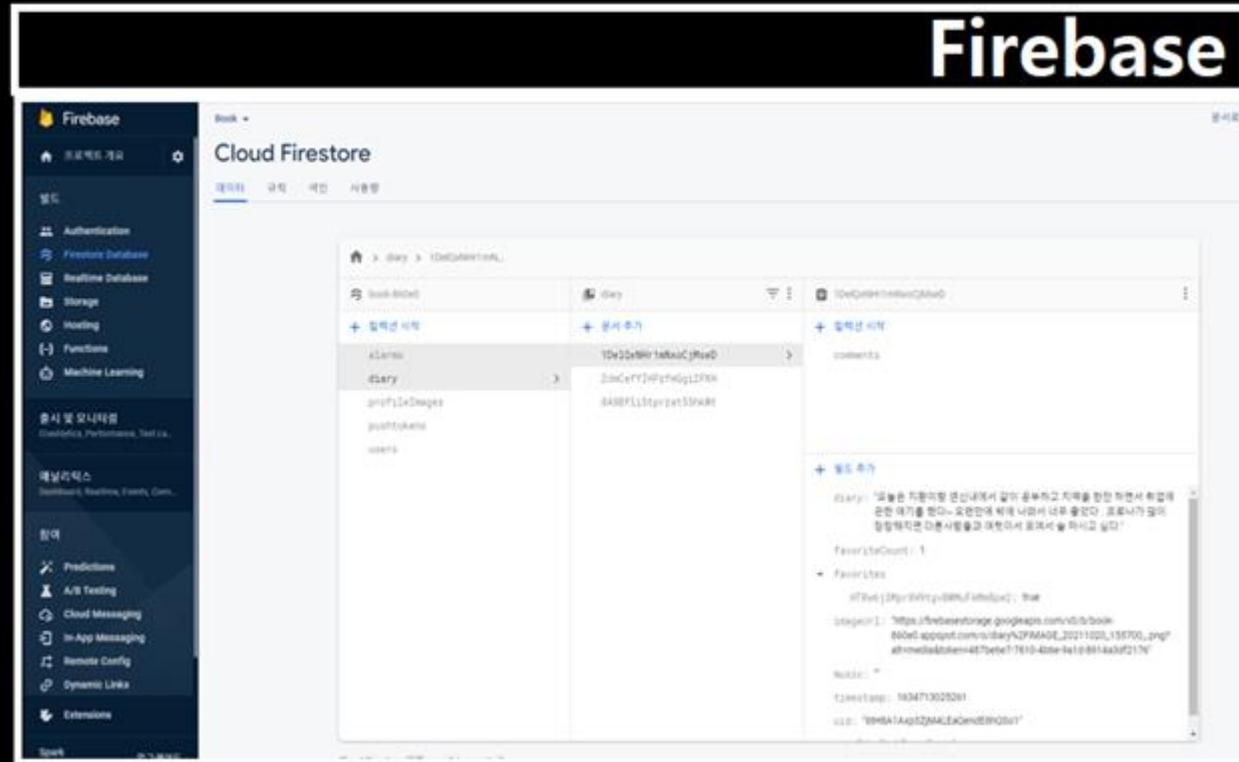
[개발 내용 및 시스템 운영 3/9]

Diary 작성

```
class AddDiaryActivity : AppCompatActivity() {  
    val binding by lazy{ActivityAddDiaryBinding.inflate(layoutInflater)}  
    var PICK_IMAGE_FROM_ALBUM = 0  
    var storage : FirebaseStorage? = null  
    var photoUri : Uri? = null  
    var auth : FirebaseAuth? = null  
    var firestore : FirebaseFirestore? = null  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(binding.root)  
  
        storage = FirebaseStorage.getInstance()  
        auth = FirebaseAuth.getInstance()  
        firestore = FirebaseFirestore.getInstance()  
  
        var photoPickerIntent = Intent(Intent.ACTION_PICK)  
        photoPickerIntent.type = "image/*"  
        startActivityForResult(photoPickerIntent, PICK_IMAGE_FROM_ALBUM)  
  
        binding.addDiaryBtnUpload.setOnClickListener {  
            contentUpload()  
        }  
    }  
}
```



[개발 내용 및 시스템 운영 4/9]



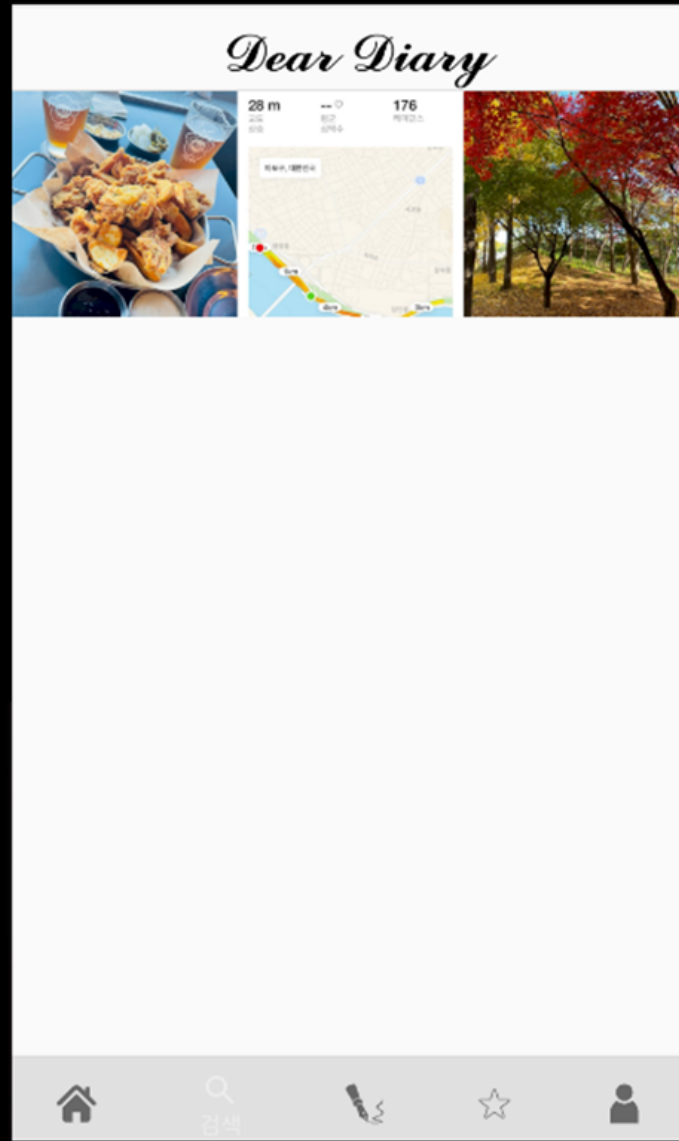
[개발 내용 및 시스템 운영 5/9]

Grid

```
class GridFragment : Fragment(){
    var firestore : FirebaseFirestore? = null
    var fragmentManager : FragmentManager? = null
    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        var fragmentManager = fragmentManager
        var fragmentManager = fragmentManager
        var fragmentManager = fragmentManager
        var fragmentManager = fragmentManager
        return fragmentManager
    }
    @SuppressWarnings("NotifyDataSetChanged")
    inner class UserFragmentRecyclerViewAdapter : RecyclerView.Adapter<RecyclerView.ViewHolder>() {
        var contentDTOS: ArrayList<ContentDTO> = arrayListOf()

        init {
            firestore?.collection("diary")?.addSnapshotListener { querySnapshot, firebaseFirestore ->
                //Some times, This code return null of querySnapshot when it signout
                if (querySnapshot == null) return@addSnapshotListener

                //Get data
                for (snapshot in querySnapshot.documents) {
                    contentDTOS.add(snapshot.toObject(ContentDTO::class.java!!))
                }
                notifyDataSetChanged()
            }
        }
    }
}
```



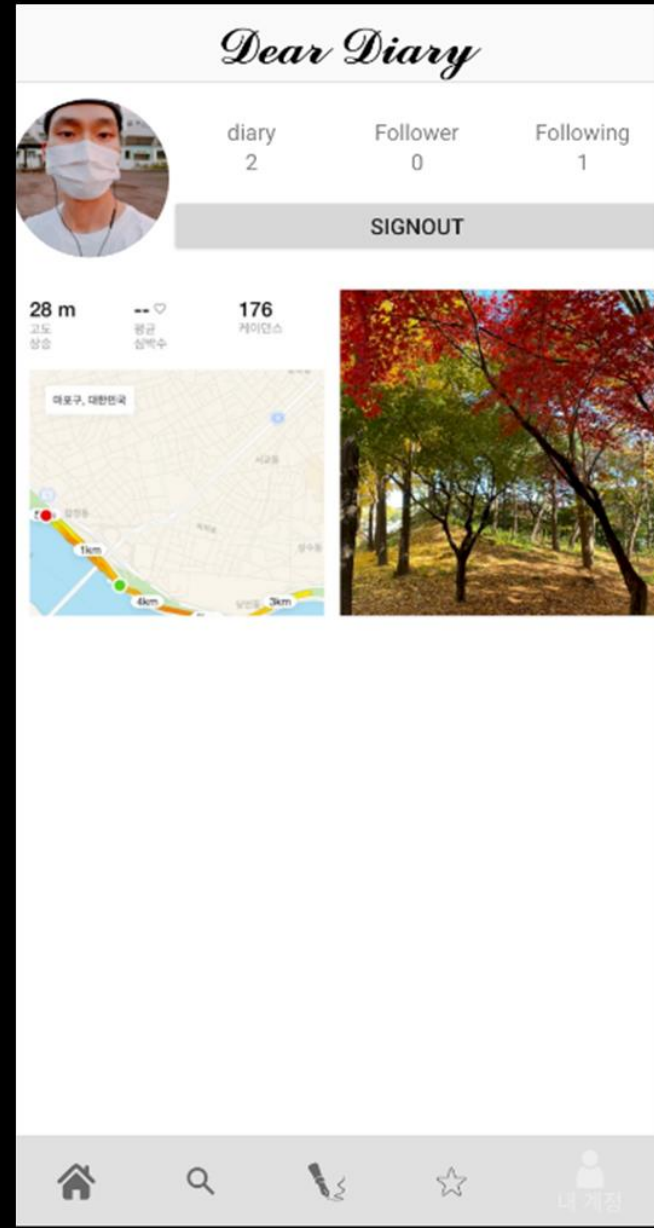
[개발 내용 및 시스템 운영 6/9]

```
User
fragmentView =
    LayoutInflater.from(activity).inflate(R.layout.fragment_user, container, false)
uid = arguments?.getString("destinationUid")
firestore = FirebaseFirestore.getInstance()
auth = FirebaseAuth.getInstance()
currentUserUid = auth?.currentUser?.uid

if (uid == currentUserUid) {
    // 내 페이지
    fragmentView?.account_btn_follow_signout?.text = getString(R.string.signout)
    fragmentView?.account_btn_follow_signout?.setOnClickListener {
        activity?.finish()
        startActivity(Intent(activity, LoginActivity::class.java))
        auth?.signOut()
    }
} else {
    fragmentView?.account_btn_follow_signout?.text = "팔로우"
    var mainactivity = (activity as MainActivity)
    mainactivity?.toolbar_username?.text = arguments?.getString("userId")
    mainactivity?.toolbar_btn_back?.setOnClickListener {
        mainactivity.bottom_navigation.selectedItemId = R.id.action_home
    }
    mainactivity?.toolbar_title_image?.visibility = View.GONE
    mainactivity?.toolbar_username?.visibility = View.VISIBLE
    mainactivity?.toolbar_btn_back?.visibility = View.VISIBLE

    fragmentView?.account_btn_follow_signout?.setOnClickListener {
        requestFollow()
    }
}

fragmentView?.account_recyclerview?.adapter = UserFragmentRecyclerViewAdapter()
fragmentView?.account_recyclerview?.layoutManager = GridLayoutManager(requireActivity(), 2)
```



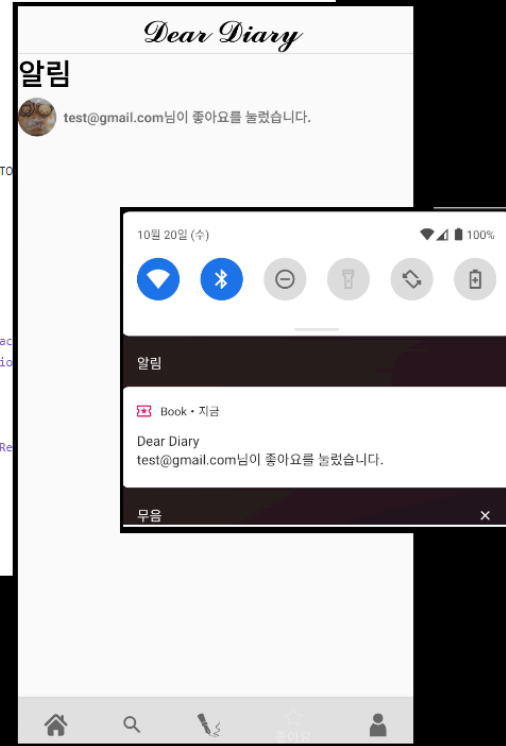
[개발 내용 및 시스템 운영 7/9]

alarm(view)

```
override fun onBindViewHolder(holder: RecyclerView.ViewHolder, position: Int) {  
    var view = holder.itemView  
  
    FirebaseFirestore.getInstance().collection("profileImages").document(alarmDTOList[position].uid!).get()  
        .addOnCompleteListener { task ->  
            if(task.isSuccessful){  
                val url = task.result!!["image"]  
                Glide.with(view.context).load(url).apply(RequestOptions().circleCrop()).into(view.commentviewitem_imageview_profile)  
            }  
        }  
  
    when(alarmDTOList[position].kind){  
        0 -> {  
            val str_0 = alarmDTOList[position].userId + getString(R.string.alarm_favorite)  
            view.commentviewitem_textview_profile.text = str_0  
        }  
        1 -> {  
            val str_0 = alarmDTOList[position].userId + " " + getString(R.string.alarm_comment)  
            view.commentviewitem_textview_profile.text = str_0  
        }  
        2 -> {  
            val str_0 = alarmDTOList[position].userId + " " + getString(R.string.alarm_follow)  
            view.commentviewitem_textview_profile.text = str_0  
        }  
    }  
    view.commentviewitem_textview_comment.visibility = View.INVISIBLE  
}
```

alarm(send)

```
fun sendMessage(destinationUid : String, title : String, message : String){  
    FirebaseFirestore.getInstance().collection("pushtokens").document(destinationUid).get().addOnCompleteListener {  
        task ->  
            if(task.isSuccessful){  
                var token = task.result?.get("pushToken").toString()  
                var pushDTO = PushDTO()  
                pushDTO.to = token  
                pushDTO.notification.title = title  
                pushDTO.notification.body = message  
  
                var body = RequestBody.create(JSON, gson?.toJson(pushDTO))  
                var request = Request.Builder()  
                    .addHeader("Content-Type", "application/json")  
                    .addHeader("Authorization", "key="+serverKey)  
                    .url(url)  
                    .post(body)  
                    .build()  
  
                okhttpClient?.newCall(request)?.enqueue(object : Callback {  
                    override fun onFailure(call: Call, e: IOException) {  
                        }  
                    override fun onResponse(call: Call, response: Response) {  
                        println(response.body()?.string())  
                    }  
                })  
            }  
    }  
}
```



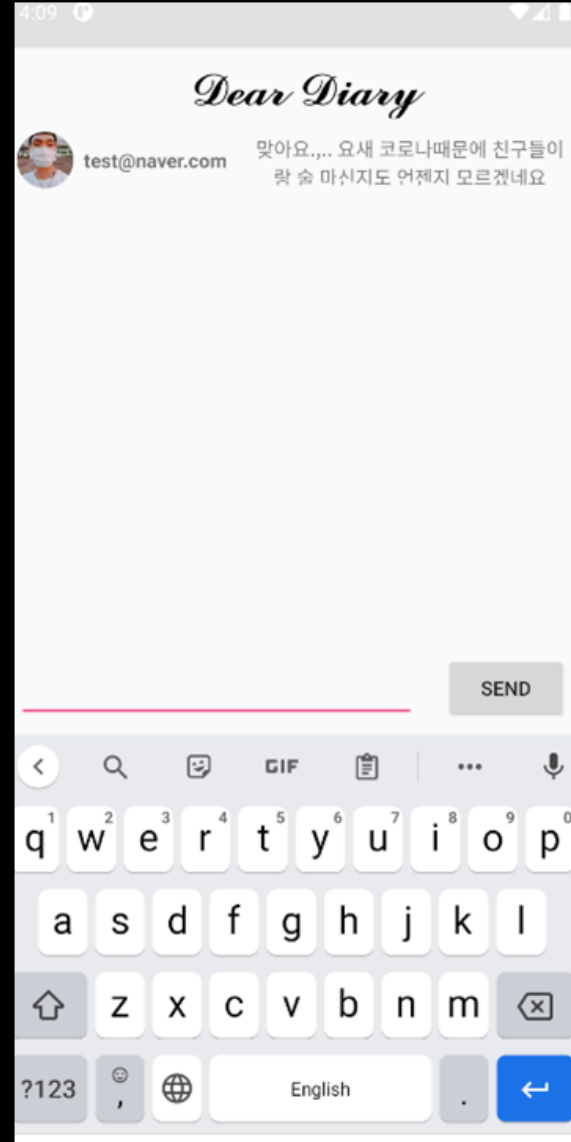
[개발 내용 및 시스템 운영 8/9]

🏠 > alarms > 97W27YAUUMS...		
📶 book-860e0	📁 alarms	📄 97W27YAUUMSYogKuke0F
+ 컬렉션 시작	+ 문서 추가	+ 컬렉션 시작
alarms >	97W27YAUUMSYogKuke0F >	+ 필드 추가
diary	Xz0LkgKEtDUmpjzbt3uw	destinationUid: "6tH8A1Axp3ZjM4LEaQendE8hQSo1"
profileImages	dd27fUXhC8UuDWhRJekK	kind: 1
pushtokens		message: "맞아요... 요새 코로나때문에 친구들이랑 술 마신지도 언젠지 모르겠네요"
users		timestamp: 1634717272580
		uid: "HTRw6j3MprXH9tgvBWMuFAMm8pe2"
		userId: "test@naver.com"

[개발 내용 및 시스템 운영 9/9]

Comment

```
class CommentActivity : AppCompatActivity() {  
    val binding by lazy { ActivityCommentBinding.inflate(layoutInflater) }  
    var destinationUid: String? = null  
    var contentUid: String? = null  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(binding.root)  
  
        contentUid = intent.getStringExtra("contentUid")  
        destinationUid = intent.getStringExtra("destinationUid")  
  
        binding.commentRecyclerView.adapter = CommentRecyclerViewAdatper()  
        binding.commentRecyclerView.layoutManager = LinearLayoutManager(this)  
  
        binding.commentBtnSend.setOnClickListener {  
            var comment = ContentDTO.Comment()  
            comment.userId = FirebaseAuth.getInstance().currentUser?.email  
            comment.uid = FirebaseAuth.getInstance().currentUser?.uid  
            comment.comment = binding.commentEditMessage.text.toString()  
            comment.timestamp = System.currentTimeMillis()  
  
            FirebaseFirestore.getInstance().collection("diary").document(contentUid!!)  
                .collection("comments").document().set(comment)  
  
            commentAlarm(destinationUid!!, binding.commentEditMessage.text.toString())  
            binding.commentEditMessage.setText("")  
        }  
    }  
}
```



[결론 및 기대효과]

□ 결론

- Diary 형태의 SNS 구축
- 알림 기능을 사용한 실시간 나에 대한 정보 알림 (좋아요, 댓글, 팔로우)
- 다양한 소셜 로그인을 통한 빠른 회원가입 가능

□ 기대 효과

- 다른 SNS와 차별점을 두어 이용자들에게 조금 더 친근감 있는 주제를 제공함으로써 편하고 솔직한 개인적인 일상을 공유할 수 있음
 - 사회적으로 대두되는 "SNS우울증" 등 기존 SNS의 안 좋은 인식을 개선할 수 있음
 - 게시물 작성 시 작성자의 추천 음악을 통하여 작성자의 기분을 좀 더 공감적인 유대감을 형성할 수 있음
-

감사합니다.

